FINANCE TRACKER

REPORT

Submitted by

KATHIRAVAN F **(2116220701505)**

in partial fulfillment for the award of the

degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2025

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Project titled "**FINANCE TRACKER**" is the Bonafide work of **"KATHIRAVAN F (2116220701505)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. N. Durai Murugan

Associate Professor,

Department of Computer Science and Engineering,

Rajalakshmi Engineering College, Chennai-602 105.

Submitted to Project Viva-Voce Examination held on _________________

Internal Examiner **External Examiner**

ABSTRACT

n today's fast-paced world, effective personal finance management is essential. This project presents the development of an Finance Tracker mobile application using Kotlin in Android Studio, designed to help users monitor and manage their daily expenses with ease and efficiency. The application aims to provide a simple, user-friendly interface for tracking income and expenditures, categorizing transactions, setting budgets, and visualizing financial trends over time.

The app allows users to input income and expense entries, which are stored locally using Room Database, ensuring data persistence even when offline. Transactions can be categorized (e.g., food, travel, shopping, utilities), and users can view summaries of their spending habits through dynamic charts and reports. The application also includes a feature to set monthly budget goals, with alerts when nearing the limit, helping users stay within their financial boundaries.

Kotlin, as the modern preferred language for Android development, offers concise syntax, enhanced safety, and improved performance. The use of Android Architecture Components such as ViewModel, LiveData, and Data Binding ensures a clean, maintainable, and scalable codebase. Material Design is implemented to enhance the visual aesthetics and user experience of the app.

The Finance Tracker app not only aids in better money management but also encourages users to build healthy financial habits. By leveraging mobile technology and modern Android development practices, this project demonstrates how digital tools can empower individuals to take control of their financial well-being.

This application can be further enhanced with features like cloud backup, multi-device synchronization, currency conversion, and AI-based spending analysis. Overall, the Finance Tracker is a practical and scalable solution for efficient personal finance tracking.

ACKNOWLEDGMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We are very glad to thank our Project Coordinator, **Dr. N. Durai Murugan** Associate Professor Department of Computer Science and Engineering for his useful tips during our review to build our project.

TABLE OF CONTENTS

INTRODUCTION

General

The Finance Tracker is a mobile application developed using Kotlin in Android Studio, aimed at helping users manage and track their daily financial transactions effectively. With growing expenses and complex financial activities, individuals often find it difficult to monitor their spending habits. This app provides a simple and intuitive platform for recording incomes and expenses, categorizing transactions, and analyzing financial data.

Users can add new transactions, assign categories (such as food, rent, entertainment, travel, etc.), and view detailed summaries through charts and lists. The app utilizes Room Database for offline data storage, ensuring that all entries are securely saved on the user's device. Android Jetpack components like LiveData, ViewModel, and Data Binding are used to maintain a responsive and maintainable architecture.

The main objective of this project is to promote better financial planning by giving users real-time insights into their spending patterns. The clean and modern Material Design interface makes the app easy to navigate and visually appealing.

This project not only demonstrates the practical use of Kotlin for Android development but also highlights how mobile applications can positively impact daily life by encouraging responsible financial behavior.

Objective

The primary objective of the **Finance Tracker** project is to develop a mobile application using **Kotlin** in **Android Studio** that enables users to effectively manage their personal finances by tracking income and expenditures in a structured and user-friendly manner.

This application is designed to assist users in maintaining financial discipline by allowing them to record every financial transaction, categorize it, and generate summaries or visual insights into their spending habits. By giving users a clear picture of where and how their money is spent, the app promotes better budgeting and responsible money management.

Key objectives of the project include:

Ease of Use: Develop a simple and intuitive interface that enables users to quickly log their daily expenses and income without complexity.

Categorization: Provide predefined and customizable categories for transactions to help users organize their financial data.

Data Persistence: Use Room Database to store transaction records locally so users can access data even when offline.

Budget Management: Allow users to set monthly budget limits and receive notifications when they approach or exceed them.

Data Visualization: Implement graphs and charts to present financial summaries, such as category-wise expenses and monthly trends.

Modern Architecture: Use Android Jetpack components (ViewModel, LiveData, Data Binding) to ensure the application is robust, maintainable, and scalable.

The ultimate goal is to help users become more aware of their financial habits, reduce unnecessary spending, and make informed decisions to improve their financial health.

Existing System

In the current digital landscape, several expense tracking solutions are available in the form of mobile applications and web platforms. These systems aim to assist users in managing their personal finances by recording incomes, tracking expenses, and generating reports. Some of the most popular existing systems include apps like **Mint**, **Money Manager**, **Spendee**, and **Wallet**. These applications offer features such as bank account synchronization, budget tracking, expense categorization, and financial goal setting.

However, most of the existing systems come with certain limitations. Many require internet connectivity and cloud-based accounts, raising concerns about **data privacy and security**. Additionally, some advanced features in these applications are locked behind **premium subscriptions**, making them inaccessible to users who seek a free, full-featured tool. Some applications are also overly complex for users who want a straightforward solution for daily expense tracking.

Another issue with existing systems is the **lack of customization**. Users may not be able to add personalized categories or manually control how their data is displayed. Furthermore, many of these apps are developed using older Android development technologies, resulting in slower performance or poor compatibility with modern Android devices.

Due to these limitations, there is a need for a lightweight, offline-capable, and customizable expense tracker. Our proposed application addresses these gaps by offering a clean, user-friendly interface, offline data storage using Room Database, and the flexibility to manage personal finances without requiring internet or premium upgrades.

2. PROPOSED SYSTEM

2.1 General

The proposed system is a mobile-based Finance Tracker application developed using Kotlin in Android Studio, aimed at providing users with a simple, efficient, and offline-capable solution for managing their personal finances. Unlike many existing applications that depend heavily on internet connectivity or come with limited free features, this system is designed to offer complete control and privacy to users by functioning entirely offline and storing data locally using the Room Database.

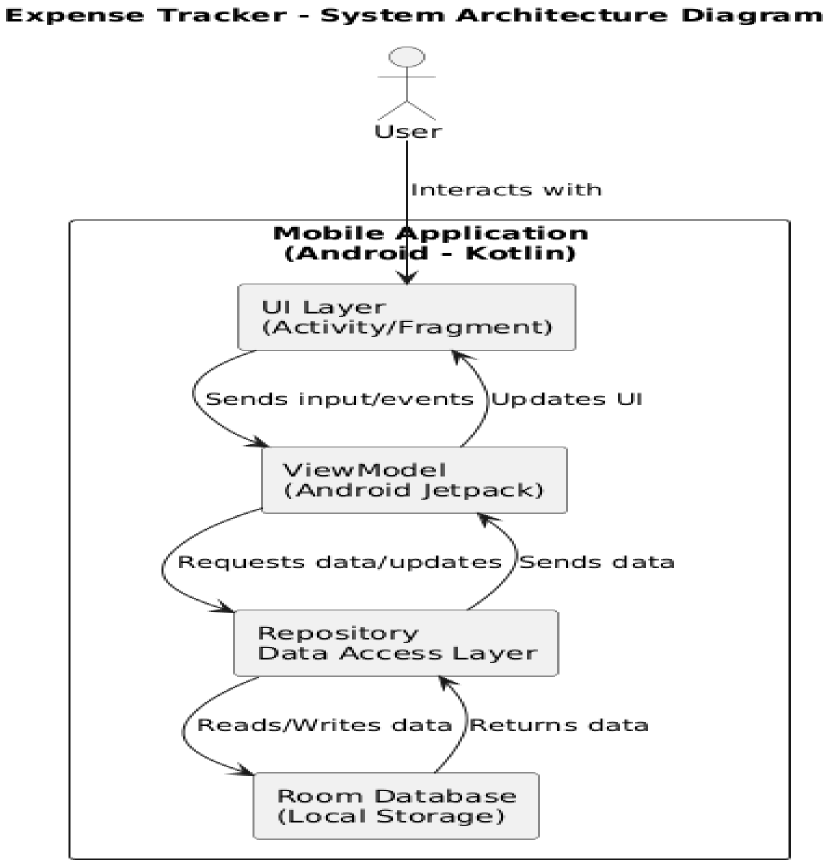
The core idea is to develop a lightweight and intuitive mobile application that enables users to log their daily income and expenses, categorize transactions, and generate insightful financial summaries through charts and lists. The app allows users to set monthly budgets and get alerts when they are close to exceeding their limits, thus promoting better financial discipline.

To ensure a modern and robust architecture, the application uses Android Jetpack components like View Model, Live Data, and Data Binding, which enhance performance and maintainability. The app follows Material Design guidelines, offering a clean, responsive, and visually appealing interface.

This system provides users with a personalized experience by allowing custom categories and a user-friendly dashboard for financial overviews. It is especially suitable for individuals, students, and professionals who prefer an easy-to-use app without unnecessary complexity or subscription barriers.

Overall, the proposed Expense Tracker app aims to address the gaps in current systems by delivering a flexible, secure, and fully functional mobile solution for effective financial management.

2.2 System Architecture



2.3 Development Environment

2.3.1 Hardware Requirements

The hardware specifications could be used as a basis for a contract for the implementation of the system. This therefore should be a full, full description of the whole system. It is mostly used as a basis for system design by the software engineers.

Table 3.3.1 Hardware Requirements

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i3
RAM	4 GB RAM
POWER SUPPLY	+5V power supply

Software Requirements

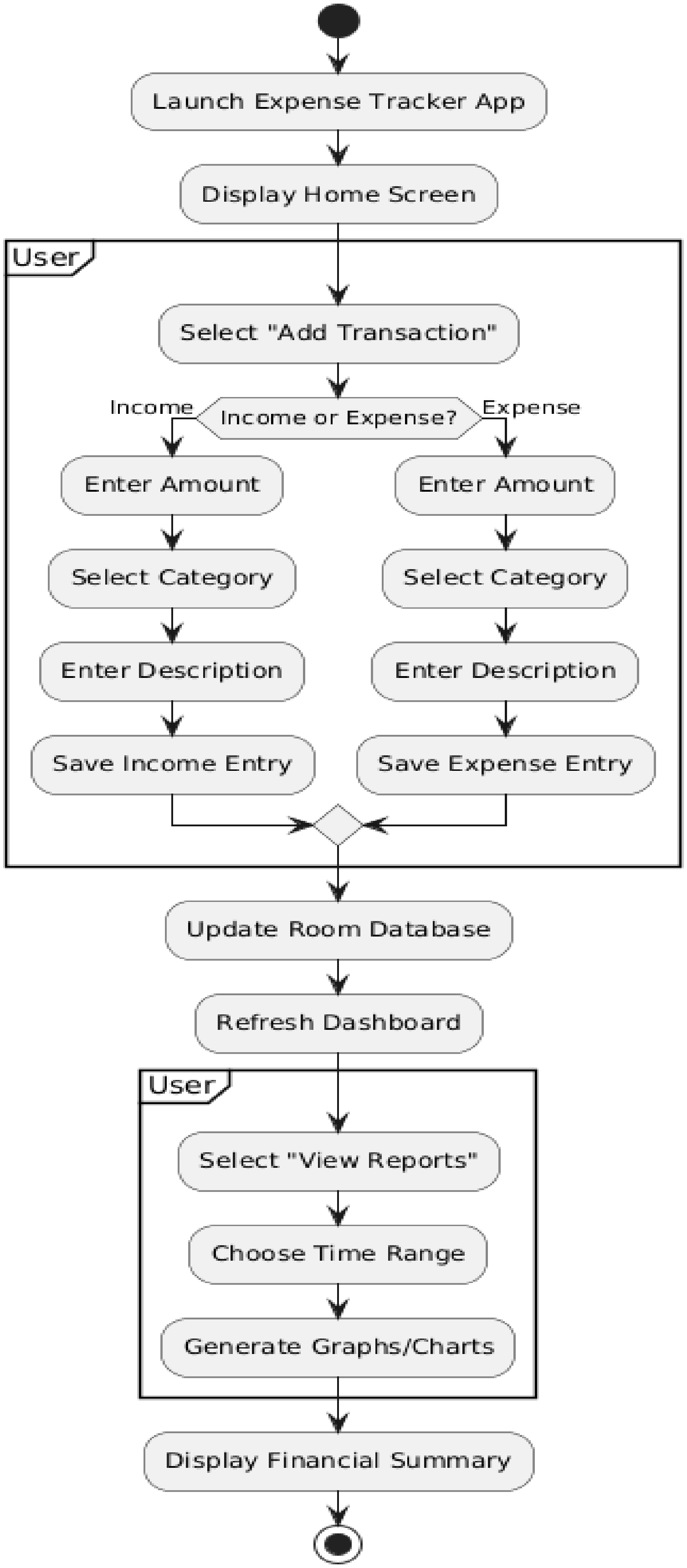
The software requirements paper contains the system specs. This is a list of things which the system should do, in contrast from the way in which it should do things. The software requirements are used to base the requirements. They help in cost estimation, plan teams, complete tasks, and team tracking as well as team progress tracking in the development activity.

Table 3.2 Software Requirements

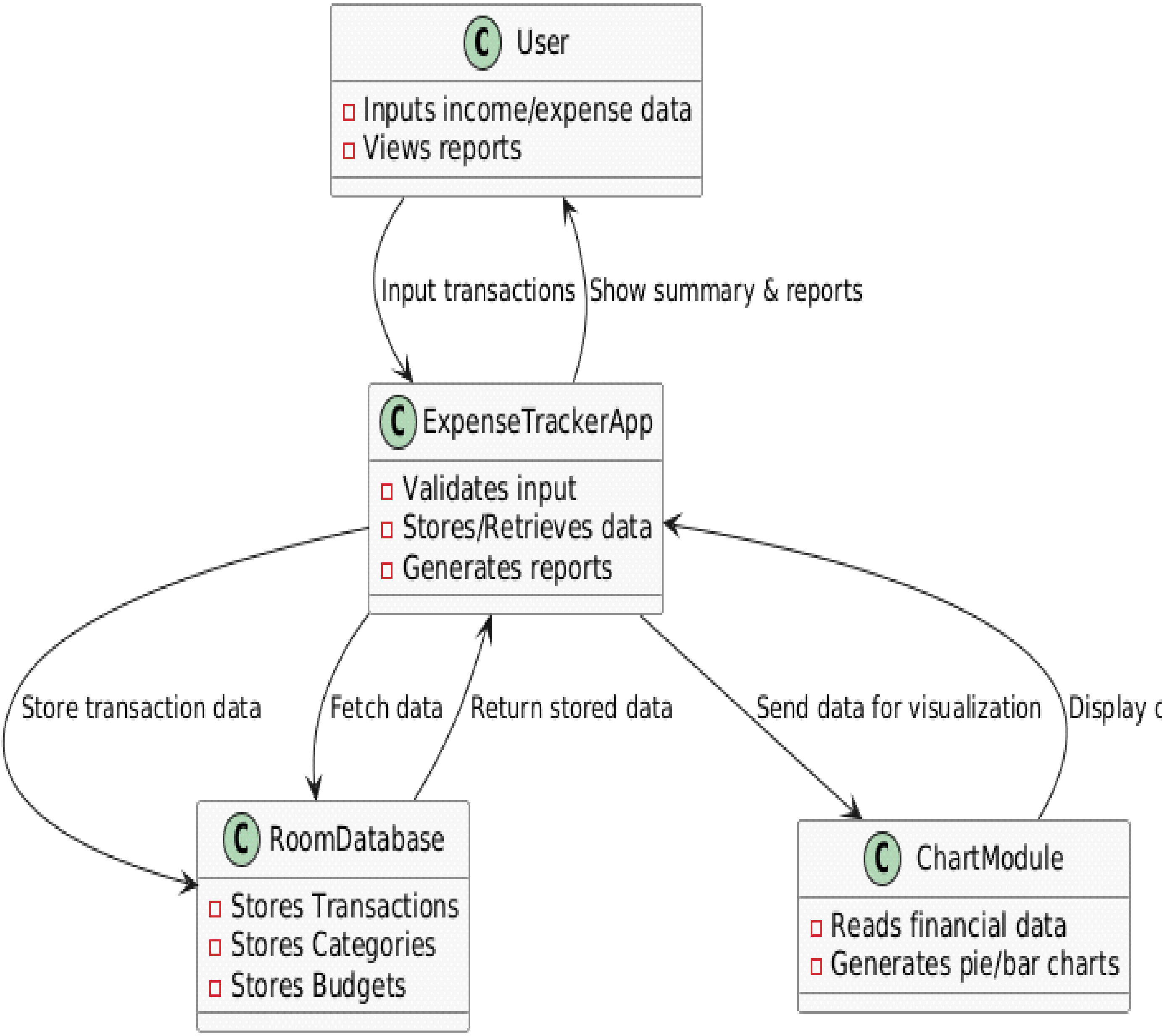
COMPONENTS	SPECIFICATION
Operating System	Windows 11 or higher
Frontend	kotlin
Backend	Kotlin

Design of the entire system

Activity Diagram


Launch Expense Tracker App
Display Home Screen
User
Select "Add Transaction"
Income or Expense?
Income
Expense
Enter Amount
Select Category
Enter Description
Save Income Entry
Enter Amount
Select Category
Enter Description
Save Expense Entry
Update Room Database
Refresh Dashboard
User
Select "View Reports"
Choose Time Range
Generate Graphs/Charts
Display Financial Summary

2.4.2 Data Flow Diagram



Statistical Analysis

Statistical analysis plays a crucial role in understanding user spending patterns, evaluating the effectiveness of the Expense Tracker, and enhancing the overall functionality of the application. In this project, the primary aim of the statistical analysis is to help users make informed financial decisions based on historical transaction data.

The Expense Tracker app collects data in the form of user transactions, categorized as either income or expenses. Each transaction entry includes a date, amount, category, and optional notes. This data is stored locally using Room Database and is processed periodically to generate meaningful statistical summaries.

Key statistical parameters computed within the app include:

Total Income and Total Expenses over a selected time period (daily, weekly, monthly).

Category-wise Expense Breakdown, showing how much was spent in each category (e.g., Food, Travel, Utilities).

Budget Utilization Percentage, which calculates how much of a set budget has been used.

Spending Trends, which show the increase or decrease in expenses over time through line or bar charts.

Average Daily/Monthly Spending, helping users compare actual spending with desired limits.

These statistics are visualized using graphical components such as pie charts and bar graphs, which provide users with an intuitive understanding of their financial status. The use of these visual tools not only aids in quick decision-making but also helps identify problem areas, such as overspending in specific categories.

Furthermore, the app may use **basic predictive statistics**, such as estimating next month's expenditure based on past trends, which can assist in setting realistic budgets. Outlier detection can also highlight unusually high expenses, alerting the user to review those entries.

Statistical analysis is implemented using Kotlin-based logic combined with appropriate libraries for rendering charts and graphs (e.g., MPAndroidChart). This ensures that the app remains lightweight yet informative.

Overall, statistical analysis within the Expense Tracker app adds significant value by transforming raw financial data into actionable insights, empowering users to manage their finances smarter and more efficiently.

IMPLEMENTATIONS AND RESULTS

3.1. IMPLEMENTATION

□ Project **Setup**: Start by creating a new project in Android Studio using Kotlin. Choose an empty activity and ensure Kotlin is selected as the programming language.

□ User **Interface Design**: Design two main screens for the app:

- **Main Activity**: A screen with a RecyclerView to display expenses, a button to navigate to the "Add Expense" screen, and a TextView for displaying the total expenses.
- **Add Expense Activity**: This screen includes EditText fields to input the name and amount of an expense, and a button to save the expense.

□ Expense **Model**: Define a simple Expense data class that includes the necessary attributes, like name and amount, to represent an expense.

- **Room Database**: Integrate Room for local data storage. Define an ExpenseEntity class to represent the table in the database. Create a Dao interface for operations such as inserting and fetching data, and create the Room database class.
- **RecyclerView Adapter**: Implement an adapter to display the list of expenses in the RecyclerView. The adapter will bind the expense data to the view components, such as the expense name and amount.
- **Adding and Saving Expenses**: In the AddExpenseActivity, set up logic to retrieve user input, save the expense data to the Room database, and close the activity when the user taps the save button.
- **Displaying Expenses**: In the MainActivity, fetch all expenses from the Room database, update the RecyclerView to show the list of expenses, and calculate the total amount of expenses to display at the bottom of the screen.
- **Database Operations**: Use Room’s Dao to perform CRUD operations on the expenses, like adding a new expense or fetching all expenses. Ensure that database operations are done in background threads to avoid blocking the main UI.

- **Additional Features**: Extend the app by adding more features, like editing or deleting expenses, categorizing expenses, generating reports, and implementing graphs or charts for better expense analysis.

3.2. Output and Screenshots

The **Expense Tracker** app is designed to help users track their daily expenses in a simple and efficient manner. Below is the expected output of the application and the screenshots for the different views:

1. Main Activity Screen (Home Screen)

The **Main Activity** screen displays a list of all expenses, a button to navigate to the "Add Expense" screen, and a TextView showing the total expenses. Here's how the screen will appear:

RecyclerView: Displays the list of expenses with each item showing the expense name and amount.

Add Expense Button: A button labeled "Add Expense" that navigates the user to the "Add Expense" screen.

Total Expenses: The total sum of all expenses is displayed at the bottom of the screen in a TextView.

Example Screenshot of Main Activity:

The list shows expenses like "Groceries - $50", "Utilities - $30", and so on.

The total displayed at the bottom might read "Total: $80".

2. Add Expense Activity Screen

The **Add Expense** screen allows users to input the details of a new expense. It contains the following elements:

Expense Name Input: An EditText field for entering the name of the expense (e.g., "Coffee").

Expense Amount Input: Another EditText field for entering the amount of the expense (e.g., "$5").

Save Button: A button labeled "Save Expense" that saves the entered details to the database.

Example Screenshot of Add Expense Activity:

Users can input the name ("Coffee") and the amount ("$5").
After pressing "Save Expense," the app returns to the **Main Activity**.

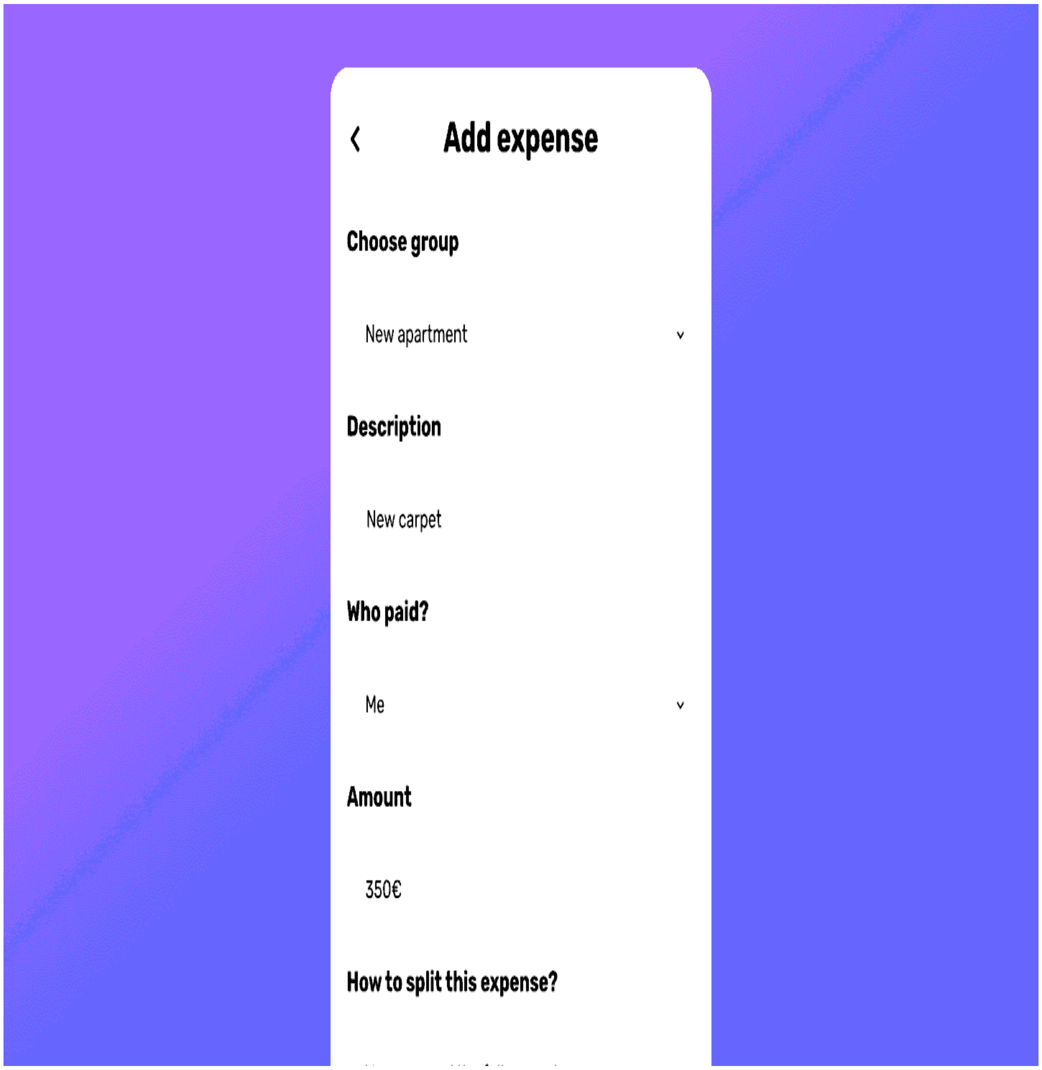
3. RecyclerView List of Expenses

The **RecyclerView** is used to display all expenses stored in the Room database. Each item in the list contains the name of the expense and the amount. It provides a clean, scrollable list of entries for easy viewing.

Example Screenshot of RecyclerView:
A list of items such as "Coffee - $5", "Dinner - $20", etc.

4. Total Expenses Calculation

The total expenses are dynamically calculated and displayed at the bottom of the **Main Activity** screen. This total reflects the sum of all expenses in the database.



Add expense

Choose group

New apartment ˅

Description

New carpet

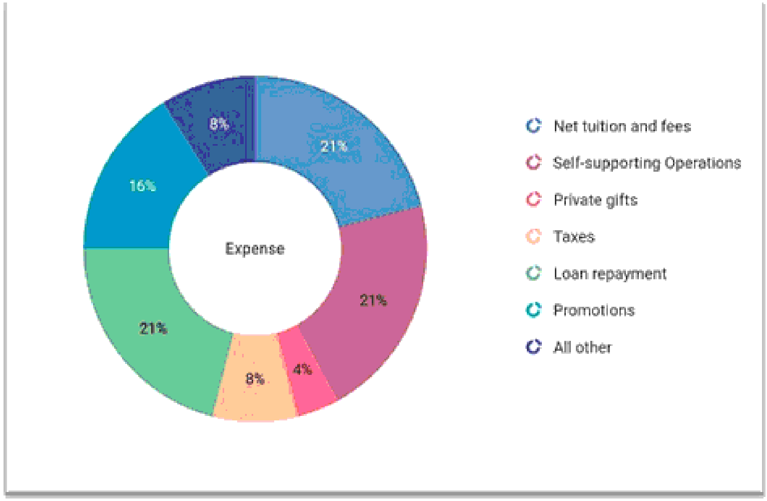
Who paid?

Me ˅

Amount

350€

How to split this expense?



CONCLUSION AND FUTURE ENHANCEMENT

4.1. Conclusion

The Finance Tracker app developed using Kotlin and Android Studio provides a straightforward, yet highly functional solution for managing personal finances. The primary goal of the app is to allow users to add, view, and track their daily expenses in an easy and organized manner. Through this app, users can efficiently manage their spending habits, ensuring better financial control. The app is designed with simplicity in mind, focusing on essential features that are practical and easy to use. With features such as adding new expenses, viewing a comprehensive list of past expenses, and calculating the total amount spent, the app is user-friendly and ensures a smooth experience.

The app utilizes Room Database, which provides local storage for all the expense data. This ensures that even after the app is closed and reopened, users' data remains intact, providing persistence without relying on external servers. The app's RecyclerView is used to display all recorded expenses in a clear and organized list, making it easy for users to scroll through their expenses at any time. Each entry includes the name of the expense and its corresponding amount, with a dynamically calculated total shown at the bottom of the screen.

The total expenses are automatically updated every time a new expense is added, which helps users track their spending habits in real-time. The app eliminates the need for manual calculation, offering users an immediate view of how much they have spent in total. This functionality helps users stay on top of their finances and avoid overspending.

In terms of functionality, the app is simple yet powerful, offering a straightforward process for adding new expenses through the "Add Expense" screen. The data entered by the user, such as the name of the expense and its amount, is stored in the Room database, which also handles retrieving and displaying this data in the main view. The app automatically calculates the total of all expenses, offering a clear picture of the user's financial situation.

While the app provides core functionalities, there is plenty of room for future enhancements. These could include adding features such as expense categories, which would allow users to group their expenses into different types like food, transportation, or entertainment. Furthermore, adding the ability to edit or delete expenses, as well as adding analytics features like visual reports (charts or graphs), would enhance the user experience. A cloud sync feature could be added to allow users to back up their data and access it across multiple devices.

Overall, this Expense Tracker app serves as an excellent foundation for managing personal finances. It combines essential features with ease of use, and with further improvements, it could become an even more powerful tool for individuals looking to take control of their financial well-being.

FUTURE ENHANCEMENT

The Finance Tracker app, while already functional, has several opportunities for future enhancements that could further improve its usability and expand its features, making it a more comprehensive financial management tool. One of the most useful upgrades would be the introduction of expense categorization. This feature would allow users to assign each expense to a specific category such as "Food," "Entertainment," "Transportation," or "Utilities." Categorizing expenses helps users gain a clearer understanding of where their money is going, which can be especially useful for budgeting purposes. This would also allow the app to provide more insightful reports and visual representations, such as pie charts or bar graphs, showing users their spending habits across different categories.

Another enhancement would be the ability to edit or delete existing expense entries. Currently, users can only add new expenses, but there may be cases where users need to update or remove expenses, whether due to input errors or changes in their financial situation. Implementing this feature would allow for greater flexibility and accuracy in managing expenses. Additionally, a recurring expenses feature could be integrated, enabling users to set up regular expenses such as rent, subscriptions, or utility bills. These expenses would automatically be added to the user's expense list on their specified schedule, saving time and ensuring that no important payments are forgotten.

To further improve the app's usefulness, adding analytics and reporting capabilities would allow users to visualize their spending patterns over time. Features like monthly or weekly reports, along with pie charts or bar graphs, would provide users with a detailed overview of their finances. This could be particularly beneficial for those who want to track their spending habits and identify areas where they could cut back or save more. Additionally, users could benefit from the introduction of a budgeting feature. This would enable users to set a monthly or weekly budget for each category and receive notifications when they approach or exceed their spending limits. It would help users stay within their financial goals and avoid overspending.

Another valuable future enhancement would be cloud syncing. By enabling cloud backup, users could access their expense data across multiple devices and ensure their information is safe from device loss or damage. Multi-currency support could also be added to cater to international users, allowing them to track expenses in different currencies and automatically convert them to their default currency using real-time exchange rates. Lastly, adding expense reminders would help users stay on top of important recurring payments, such as bills or subscriptions, ensuring they don't miss due dates and incur late fees.

These future enhancements would significantly improve the app's functionality, making it not just a simple expense tracker but a comprehensive personal finance management tool.